PRIVATE

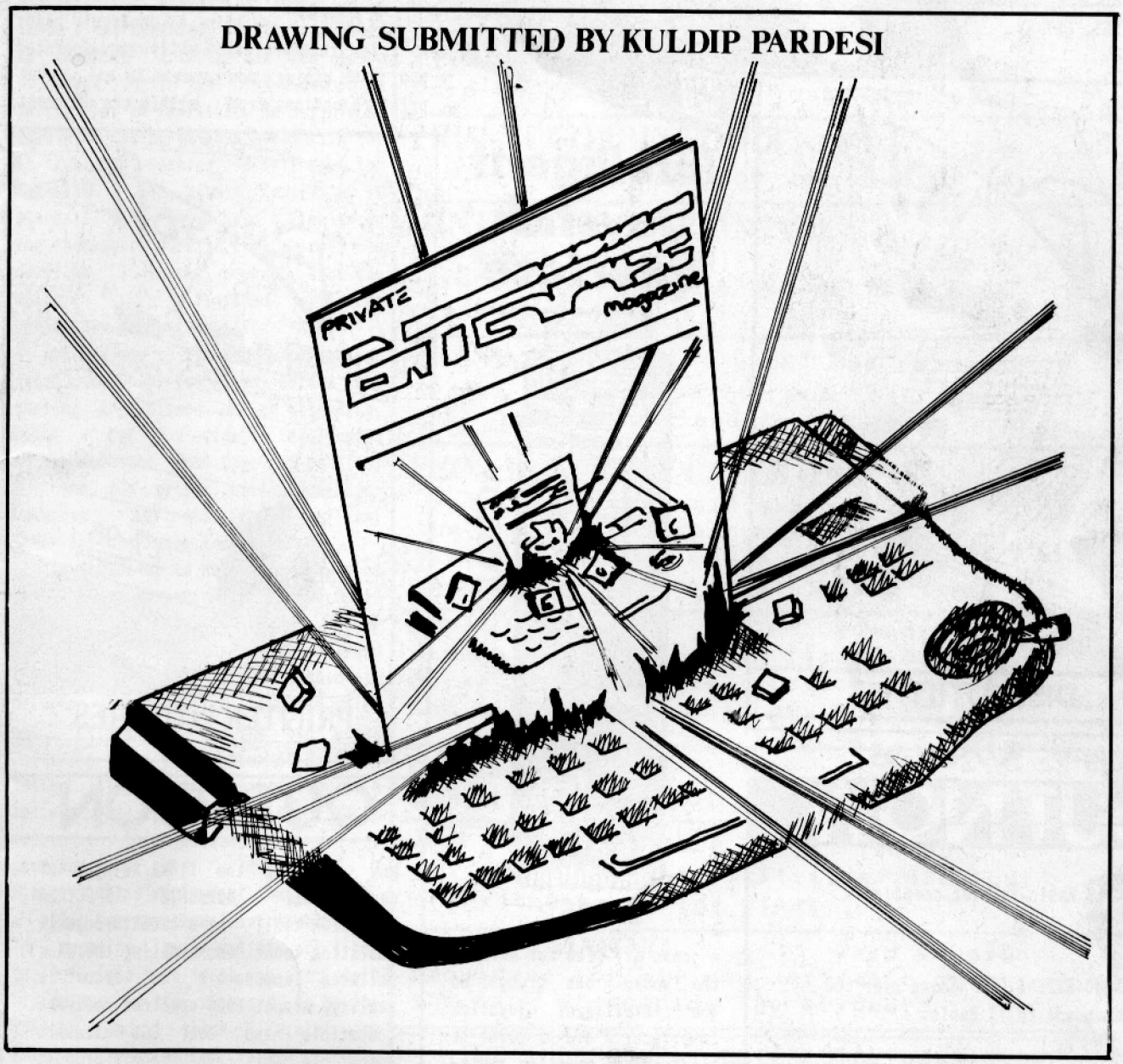
ENIGMA

magazine.

JANUARY/ FEBRUARY
1987

ISSUE 8
An I.E.U.G publication

DRAWING SUBMITTED BY KULDIP PARDESI



EXOS FUNCTION CALLS PART ONE

SOUND ADVICE | Hints & Tips Extravaganza | KEYBOARD SCANNING | CP/M'S LATEST | NEWS & VIEWS

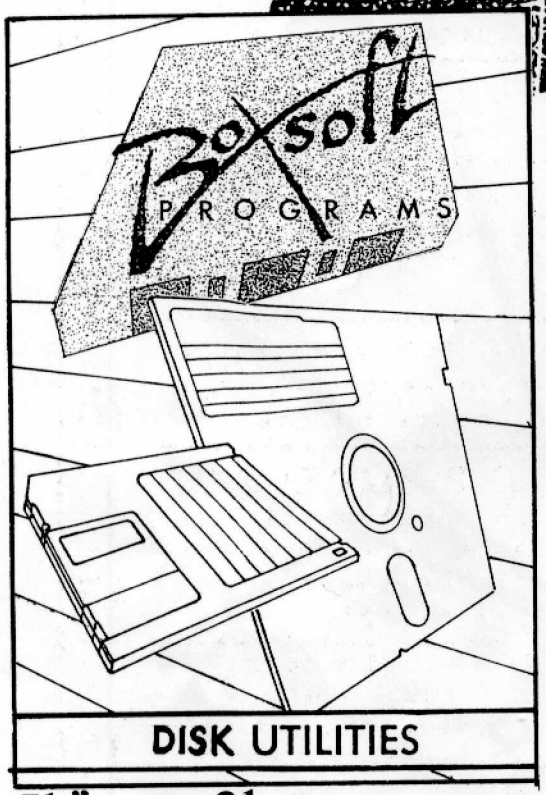
Disc Utilities

All disk users at some stage desire the ability to have direct access to the data on their disks, be it for salvaging corrupted files or just altering programs without having to first load them in. DISK UTILITIES allows you all this and much more. A powerfull disk sector editor is just one of the CP/M tools included on this utilities disk. Others include DISK, STATUS, CONVERT, DUMP, FKEY, SETKEY and WIPE.

Boxsoft PRESENTS

Printer Utilities

Are you always waiting on your printer? Do your documents look drab and boring? Then we perscribe PRINTER UTILITIES. With this indispensable tool you can now carry on working whilst the new printer driver continues to print out up to 14K of text and graphics. PRINTER UTILITIES also allows you to place control codes within a document ready for interpretation whilst being printed, and thus giving you access to all your printers options from within the document.



DISK UTILITIES

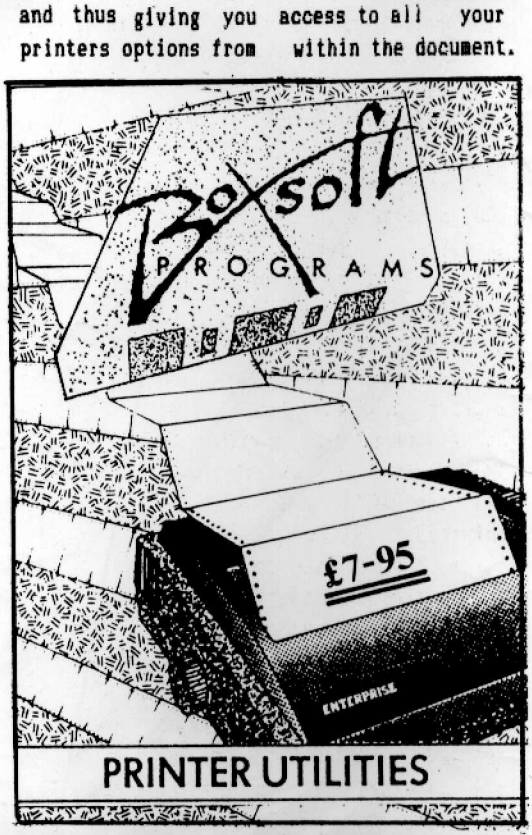
5¼" £10-95 3½ £11-95

Intelligent Joystick Interface

ACTUAL SIZE

£9-95

A spin off from our work on the mouse has brought us the intelligent joystick interface. A clever piece of electronics enabling auto-repeat, twin fire buttons and the ability to use more sophisticated products, such as trackerballs and mice.



PRINTER UTILITIES

OUT NOW

ZZZIP IS Basic integer compiler

Price £12.95

IS BASIC EXTENSIONS Adding over 100 new commands to IS Basic.

Price £7.95

SCREEN UTILITIES Scrsave ,Scrload, Scrcopy and Scrdump all your graphics.

Price £5.95

SOEV Sound and Envelope defining utility with auto program generation.

Price £5.95

PRINTER CABLES Boxsoft's own brand with full polarization slots.

Price £10.00

OUT SOON

Now reaching the final stages of development, BOXSOFTS SPECTRUM EMULATION UNIT is a breakthrough in emulation techniques enabling the once believed impossible to become a reality, almost 100% spectrum software compatible and over 60% hardware compatible. This revolutionary product will open up the Enterprise to probably the largest selection of games software in the UK today. Improved design has also enabled us to cut down on components thus reducing the price well into everybodys range. This produce we believe will turn Enterprise computing up side down.

PAINTBOX and MOUSE

A full functioned art package and mouse with textured fill and spray being just two of it's features.

Boxsoft

12 WHITEGATES, 100 STATION RD
NEW BARNET HERTS EN5 1QB

Editorial

Yes, I know.... the unbelievable has happened. An issue has been produced approximately on time ! If you look on the back cover of Issue 7, you will see the immortal words "Issue 8 will be available early February 1987".

... and here it is !

However, not everything promised in Issue 7 has appeared. The "Introduction to IS-BASIC" article has had to be postponed until next issue due to it having to be rewritten. An article of this kind has to be absolutely right or it achieves nothing. I felt that the material I had wasn't really up to standard, and there wasn't time to do the necessary work in time to meet our deadline. I think, however, that the article when finished will be definitely worth the wait.

A message now to everyone who received Issue 7 only two days before the meeting and missed out on attending. Don't worry, you didn't really miss much. See the News page for details, but the AGM proper will be held on Saturday April 4th. Details of that later in this issue too.

A big thumbs up to Martin Gillespie of Leyton who was quick to respond to an enquiry in Popular Computing Weekly with a reply letter which was published the following week. I was absolutely deluged with enquiries (51 phone calls and letters !) in the two weeks following the publication of Martin's letter. If something tucked away in a letters page can produce that kind of response, I think we should all get our pens and paper out !

Finally, while you've got that pen out, a plea for some more contributions to THIS magazine. The amount of work I had to put into this issue was absolutely ridiculous - I'm supposed to be the Editor, and I think I wrote more articles than I edited! Please could we have some more articles - they don't have to be literary masterpieces or theses; if you feel you have something to offer, please write in. The same few people have been writing all the articles since Issue 1 and are reaching the limit of their endurance. Many thanks to both Mike Turner and Chris Holland for their first-time efforts in this issue.

Neil Blaber

PRIVATE ENTERPRISE magazine. Jan/Feb 1987

CONTENTS...

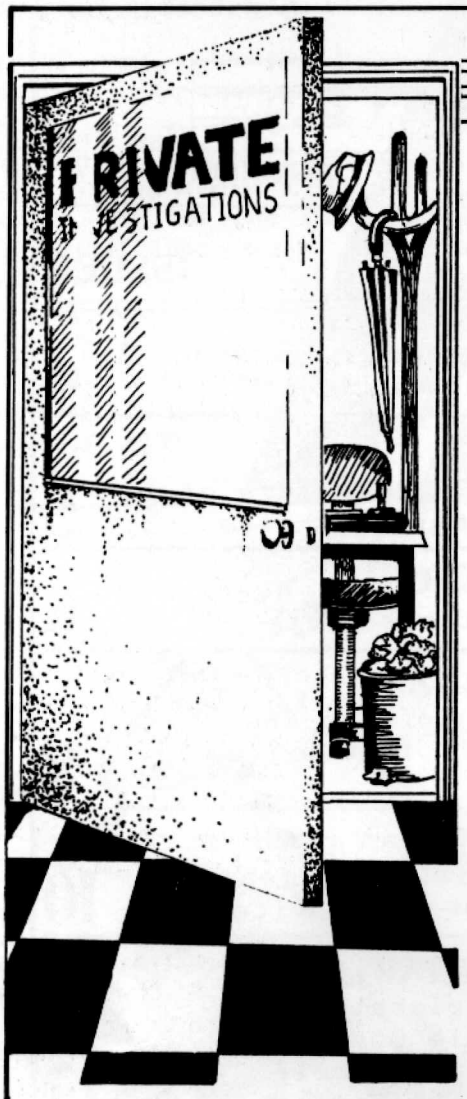
ISSUE 8

THE INDEPENDENT ENTERPRISE USER GROUP

P.O.BOX 13,
Crowborough,
East Sussex TN6 1QX

An Independent Enterprise User Group publication. Original artwork and concept MARK LISSAK, Editor NEIL BLABER, Production EAMONN O'LEARY Articles from (in alphabetical order) : CHRIS HOLLAND, DAVE RACE, GARY THOMSON and MIKE TURNER.

News Desk



The A.G.M. that wasn't

Due to the fact that most members had a grand total of two days' notice for the A.G.M., the stage was set for the biggest anti-climax in User Group history. As things turned out, it was actually quite a useful meeting, although it never actually became the A.G.M (more on that later).

An unexpected delay at the printers (due to a blatantly libellous line Neil had written into his Verran piece which had to be withdrawn) and the particularly bad postal service (second class post took about six days to arrive) saw the predicted two weeks' notice of the meeting cut to two days. This prevented a large number of members from making it down to the meeting, although the Hotline was plagued with messages of support from most of them !

After various initial problems, like not being able to get into the hall and nobody remembering to bring the coffee, the meeting started with Tim doing his sales pitch and various new bits of software being demonstrated. Unfortunately the mouse interface comitted harakiri in transit from Barnet, so there was no mouse demo. For those who hung around afterwards, there was also a tantalising glimpse of WordPro 100, the CP/M word processor Neil is currently working on.

At about 3.00pm, the meeting proper began with Neil chairing and Martin Wallace acting as Minutes Secretary. Due to the circumstances which preventing so many from attending (there were only 22 members present), Neil proposed an emergency motion that all decisions made at the meeting should be agreed in principle only, and would be ratified at an Extraordinary General Meeting to be held early in 1987. This was seconded by Martin Wallace and passed by a unanimous vote. This meeting will be held on Saturday April 4th. from 1.00pm - 5.00pm at the same venue. Various amendements were then made to the constitution, which had been circulated at the beginning of the meeting. When Agenda Item 3 was reached, "that the constitution be approved with immediate effect, and that the meeting then become the A.G.M.", this could not be passed due to the initial motion. The meeting therefore never became the A.G.M., and discussion on the other agenda items was postponed until April 4th.

Hotline gets a warm reception

The IEUG Hotline, initiated and manned exclusively by Neil from his hovel in Crowborough, is an amazing success. The phone hasn't actually stopped ringing since it was installed in December ! A good two-way flow of news, hints and tips has continued unabated well into February, by which time any novelty factor associated with having the IEUG "on-line" in the evenings should have worn off. It looks like the Hotline is destined to become a major part of User Group life... and that Neil is destined to get very sore ears ! In fact, so that he can actually get some work done (or some rest !) on Sunday afternoons, he has decided to restrict the time to evenings only, 7.00-10.00pm, 7 days a week.

IEUG forsake T.E.I. for D.I.Y.

In the wake of last Issue's slating of Verran and the adoption of TEI as the officially endorsed User Group repair company, we have now discovered that TEI weren't actually much better than Verran and have decided to provide our own repair service. This will be carried out by Tim Box, and repair enquiries should be directed to him at the Boxsoft address (found elsewhere in this magazine).

Disc Drive offer now ready

Prices have now been fixed with Cumana, and are as follows :

Single	3.5" drive	£120.00
Twin	3.5" drive	£200.00
Single	5.25" drive	£150.00
Twin	5.25" drive	£270.00
1+1	3.5" + 5.25"	£260.00

Drives include power supply, leads and moulded fused plug. Prices quoted are inclusive of VAT and courier delivery. The price of 5.25" drives fixed with Cumana are rather high, so we advise members specifically looking for 5.25" drives to look elsewhere before taking up this particular offer. Technomatic of London, for example, are selling single 5.25" drives at £120.00 each. Please send orders to "IEUG Disk Drive Offer" at the Crowborough address.

Ye P.C.W. Show Review

Time just files by, doesn't it ? It only seems like yesterday when we were cowering in the corner of the Enterprise stand at last year's PCW Show. Of course, this year circumstances were very different to those encountered in 1985. For starters, we had our own stand - just as well, seeing as Enterprise had to cry off due to certain difficulties involving the United Commercial Bank and the Inland Revenue. Also, due to Enterprise's non-appearance/existence we had a free hand to approach all and sundry in Ye Queste for Ye Surprisingly Scarce Sacred Supporting Software. And so it was that our brave stalwarts emerged on the balcony of Castle Olympia just in time to be goosed by a pervy robot... if you are not of a nervous disposition then read on !

YE CASTE IN ORDER OF APPEARANCE

Neil Blaber Gary Thomson Duncan Taylor Tim Box Andy Burnham Dave Race Andy Fitter

Chapter ye Firste - Prelude to Ye Queste

"'Twas early in ye day before the first Trade Day when our intrepid band of adventurers assembled outside the walls of Castle Olympia - Sir Spikey the Fluorescent, Sir Garfield the Diminutive and Sir Duncan the Lubricated arrived first, soon to be joined by Sir Timbox the Soft and Sir Dataquip the Hard. Entry was gained under the very noses of the somnambulent orc guards, and the first tentative steps were taken towards the Holy Stand of Zob. Alas, when discovered, the Stand had been plundered of all its riches, and all that remained were a wooden table (folded) and a plastic chair (uncomfortable). The walls had been stripped of their fine tapestries, and there was only one electric socket. A great depression did fall upon the assembled Knights, but it did not last long. A band of wandering serfs were encountered and donated two chairs and a table to the Cause when threatened with Sir Spikey's mighty hairdo. Much rejoicing was there in the Kingdom that night."

Chapter ye Seconde - Attack of the Lynx Beast

"The knights rose with the dawn and prepared to the defend their stronghold against the fearsome assault of the great Public. However, their plans were nearly thwarted by the appearance of a vast bloated monster, clouds of toxic gas emanating from the region of its armpits. The Knights charged in to despatch the creature and cleanse the Land of Evil, but were unable to get within striking range due to the pungent aroma. The creature suddenly reached within the rolls of fat and produced an amulet with the mystic runes "Lynx U.G." inscribed upon it. The Knights were sorely afraid, but Sir Garfield produced his magic book and cast the mighty spell of Ridicule upon the creature. This great spell only bounced off the monster's crusty exterior, but distracted it enough so that it noticed the picture on the old black & white telly on its stand was rolling, and it shuffled off to give it a kick. The crisis had passed, due to the quick thinking of Sir Garfield, and the knights retired to recount ballads and change their underwear.

After a suitable rest, a number of Knights did embark upon a perilous Quest of exploration, and did discover that our stand looked pretty duff in comparison to the others. It was decided that Sir Spikey, having the worst dress sense, should go forth and find some White Sticky Tape. Sir Spikey duly complied, and blundered off into the unknown with a ridiculous grin on his face. Miraculously, he managed to locate a merchant with the goods after only a few hours of searching. Taking a short cut on the return journey, he entered Castle Olympia via the back entrance. The sight which greeted his eyes upon his return did freeze him to the spot in terror. A great enchantment had been cast upon the occupants of the Castle, transforming all the computers into cuddly toys ! "Gadzooks," said Sir Spikey, who spent many a minute in deep thought before it dawned on him that he'd wandered into the Soft Toys Fair by mistake. Much michael extraction at the expense of Sir Spikey was there in the Kingdom that night.

Chapter ye Thirde - The Tin Pervert

So many traumas, and yet the brave Knights had not met the mighty Public ! Eventually, this momentous event had to occur, and so it did. Strangely reminiscent of the Trade Days last year, the Knights were besieged by a vast host of little kids, all intent on ripping the joystick off the keyboards of the display machines. Only by resorting to mindless violence were they able to prevent this, and of course all enjoyed this immensely.

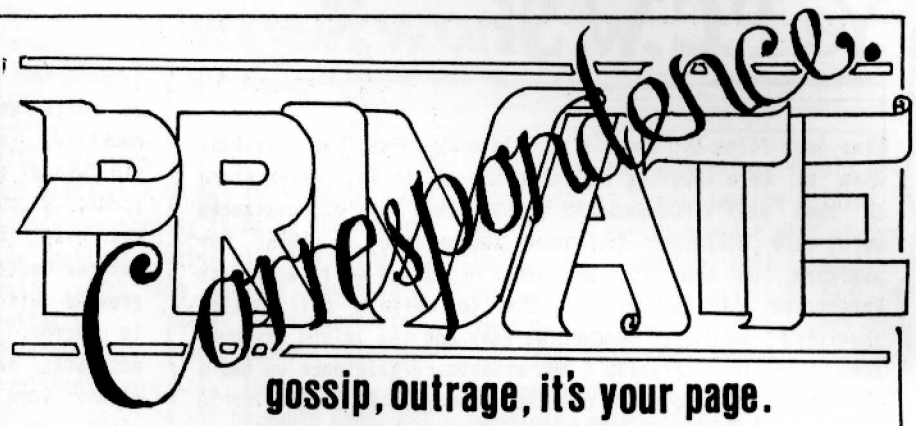
As the day progressed, strange spectres of the past were raised amongst the fortifications of the Stand. The face of Michael Shirley was observed on more than one occasion, as the immortal words "it wasn't my fault" echoed ominously overhead. Awed by this vast apparition, the Knights were sorely afraid and may finally have succumbed to the onslaught of the little kids. But Lady Luck smiled upon them and this was prevented by the appearance of an unlikely ally in the form of the sexually depraved Ariolasoft robot. This metal maniac seemed to be involved in promoting a game about pinching peoples' bottoms, as this was his major activity, with creeping up behind little girls and saying "boo" loudly coming a not-very-close second. So proficient was he at these activities that the attacking children fled in terror. Much locking of bedroom doors was there in the Kingdom that night.

Chapter ye Fourthe - Of the Six Megahertz and Slimy Pizzas

With the defeat of the initial host of infant barbarians, the Knights became bolder, especially with the arrival of reinforcements in the shape of Sir Spiderman and Sir Andy the Inebriated. First, Sir Andy revealed his Bouncing Ball which drew great crowds of onlookers. Then, Sir Timbox scrabbled beneath the Stand and produced the fantastic Six Megahertz. All six had never been observed together at the same time, and the speed of 3D Starstrike was ridiculous. Even the Great Druid Minter forgot his llamas and drifted across Olympia on a cloud of intoxicating smoke to observe the Megahertz.

Meanwhile, Sirs Spikey, Garfield and Spiderman ventured forth to gather pizzas for a hearty repast, only to discover that those on sale were the ones they didn't sell last year. Slime abounded, and Sir Spikey was almost slain by his two enraged

(CONTINUE YE QUEST PAGE 20)



Hi, it's your fun-loving, smooth talkin' and completely overworked old Editor here. The move to Sussex has prompted something of a deluge of mail, so much so that I've only managed to dig myself out in time to answer some of 'em here ! Thanks to everyone who wrote in praise of Issue 7, although the first letter this Issue isn't quite so complimentary....

"YOU ILLITERATE, SELF-INDULGENT..."

I recently received a copy of "Private .....". It has reminded me that I intended to write to you after issues 1,2,3,4,5 and 6. I put the dotted lines in the title as on the cover it is impossible to read the other word in the title. I have just quickly read through the issue and have noted nearly 60 spelling/typing errors, apart from the language which is, in many places, verging on illiteracy.

I am sure that I am not alone in having bought an Enterprise Computer in order to learn about elementary computing and especially with the objective of word processing. I find the machine a complete white elephant. I have not bought a printer as what little I have read leaves me uncertain as to whether there is a cheap daisy-wheel which I could guarantee would be compatible to just plug in and work.

I am not particularly numerate and have no knowledge of computing. Your articles all appear to be written by electronics/computing students who haven't a clue either what the layman needs nor how to communicate in plain English. The magazine is written in a sort of transatlantic jargon and is totally self-indulgent. I find it easy to see why few people attend your meetings. The appeal can only be to a tiny minority of like-minded dilettantes.

I return to one of my earlier points. The whole point of word processing is to enable text to be viewed before printing. Could anyone take you seriously when you do not even use the basic functions of the machine !!

Peter Carter
South Wirral
Cheshire

NB. I won't print the reply I sent to this letter, as it was my own personal reaction to what Mr. Carter had to say. However, I'd be interested to hear the readers' reactions to the points made here. I know all about the spelling mistakes problem - hopefully this issue shouldn't contain many, but if you find the articles incomprehensible please write and tell us - maybe a major rethink is in order.

TALES OF MEPHISTO

In Issue 7 of Private Enterprise, you wrote that the assets of Enterprise have been bought by Broadlight Ltd. of London W1. Well, I have heard that the German company Mephisto has taken over the production and sale of Enterprise and accessories. Do you know anything about this company, ie. their address ?

I don't understand why you never wrote about this, but it is obviously well known by a lot of Enterprise owners that the first Enterprise 64s do have some errors which makes the expansion port useless. I've just bought an EXDOS module (so has one of my friends) which works as it should on a newer Enterprise 128, but doesn't work on our old 64s. Once I spoke with the previous president of the Danish Users Group, "Enter", and he knew what to do about it (some resistors were wrong, as far as I remember) but I don't have his name or number any more; so I ask you - what can I do about it ? I have a little knowledge about electronics, but am able to solder.

Finally, I would like to know if it is possible to buy an upgrade of the EXOS and Basic 2.0 ? If yes, then by whom ? And the price ?

Jorgen Winther
Toftlund
DENMARK

NB. It looks like you've got some information on the fate of Enterprise GmbH, (see Issue 2 News for details) who sold the machine in Germany under the name Mephisto. They mysteriously disappeared about the time Enterprise went under, so they may have resurfaced using the name Mephisto Computers. We don't know their address, but if anyone finds out please let us know.

Oops ! Looks like a major oversight on our part about the duff machines. Personally, I didn't know that there were any faulty 64s knocking about, although the early 128s were all unable to work with the disk interface. There are some internal modifications needed to rectify this, for which we can supply instructions (Gary successfully got his machine working using these mods, so we know they work).

I'm afraid we're in a stalemate with the EXOS and Basic upgrades at the moment, but I can say things should improve within the next month or so.

TIP FOR THE TOP

I have just received Issue 7 of Private Enterprise and I think it's great, but I think it would be a better buy having a 'Playing Tips' section where people could send in tips for games. Maybe you could have a Top Ten list of software.
I would also like to know what happened to games such as Super Pipeline 2 and Cauldron. Are they ever going to be released ?

Maybe you could have a competition every issue where you could give away a game ? Competitions could have questions or a practical answer such as drawing or programming.

Here are a few playing tips of my own :

KING OF THE CASTLE - To complete the game you must get rid of the cat which blocks the exit. To do this, you must collect the charcoal and the sulphur and drop them both over the saltpetre in front of the cat. Then drop the Magic Flame onto all of this and you can say goodbye to the cat !

THE ABYSS - When coming to drop the bomb to blow up the force field, drop it from the top of the screen, then go out of the screen and back. If there had been any photon thrusters they will have disappeared.

I need some help as well - does anyone know what to do in Airwolf except blow yourself up ?

Finally, I have sent you a small picture. What do you think of it ? Could it be used as a cover drawing ?

Kuldip Pardesi
Manor Park
London E12

NB. Cheers for the artwork and the games tips - as you can see, we've actually started a Tips Page this issue. Although this will cover both programming tips and games tips (not really enough of either to warrant a page to themselves), it will definitely fill a gap in the contents of the mag. Are there any other budding artists out there willing to show their talent ??

FORTH ATTEMPT

Once upon a time I thought I knew a little of the philosophy of Forth programming, so I purchased IS-FORTH. Problem ? Well, the manual is written in pdeudo-intellectual jargonese, and reading it in an attempt to understand how to use this Forth has left me lost and confused. Can anyone help ? And can anyone define an INTEGER SIN function (not as silly as it sounds) ? What I mean is a Forth word equivalent to the BASIC line :

MOD (1000*SIN(X))

If you can answer these you'll make me a very happy man. I might even be able to produce some pretty, if not very useful, Forth routines for Home Produce (well, they say bribery will get you everywhere) !

Peter Stensones
Hull
North Humberside

NB. I'm afraid there aren't any Forth buffs amongst the magazine stalwarts, so we'll have to rely on someone out there to answer this query. I know there's at least one of you that should know the answer (Richard Mishra and Andrew Richards take note !).

DODGY EXCHANGE

Firstly let me congratulate you on producing the best yet issue of Private Enterprise; Issue 7 was a thoroughly good read. The article on the Enterprise Saga was brilliant. The news that a Spectrum emulation unit is on the way is great, but one thing I would like to know is whether it includes games that would run on the 128K Spectrum (for 128K Enterprise owners, of course).

The Software Register could have been a good idea, but you may be helping pirates get copies of new games. All they have to do is copy the original and offer it to you to swap (they keep the copy and offer the original). Are you going to continue this possibly dangerous service ?

I have a couple of questions about the IS-BASIC Extensions program on sale from Boxsoft. Is it compatible with Zzzip, ie. can it be compiled and, secondly, how is the assembler used ? Would it be worth a machine code programmer getting this program just to use the assembler ?

John Foster
Sheffield

NB. The idea for the Software Register wasn't actually mine, so I think it best that the organiser of this service answers your particular criticism:

The decision to start the register was not taken lightly; however in the end it was agreed that the need to put users who could help each other out together outweighed the slight risk of abuse by dishonest participants. It must be remembered that, as yet, there are no new games as such avalible for piracy or otherwise and this includes legitamate sale; only existing games that are not at present available at all come under this scheme. The risk of piracy as you outlined is in fact very small as the pirate would have to give up his or her hard won original in order to take part. It would be the pirate who in effect would be left with the illeagal copy and for no real financial gain. Compare this with what usually happens with pira cy and you will see it's hardly worth the effort. In answer to your final question on this matter, no, we do not (and indeed did not) intend the register to be anything more than a temporary service to get us over the software blackspot we are at present in. All those who took part will be notified one way or the other by the end of April at the latest, by this time the software situation should be resolved and the register will be closed for good.

The other query is really Boxsoft's domain, so I'll pass the buck on that one, too ...

The short answer is no. The reason for this is ZZZip was written months before the relase of IS Extensions so Peter Hiner did not know of the extra

CONTINUED ON P.21

MIKE TURNER'S HINTS

Welcome to a new page in the magazine. This spot is going to be a "hints and tips" column for members who are having trouble using the machine or playing games (Arcade, Adventure etc...) and for members who have tips for making life easier on the Enterprise.

Hi there - my name is Mike Turner and I'm the man who will be writing this action packed, fast moving, wild living death defying page. (Well... it sounds good). How they ever roped me into this I'll never know. Maybe it was something to do with my arm being forced up my back. (Just joking chaps, I'm loving this). I'm here now, armed with my trusty Enterprise, fingers nimble, mug of tea and rearing to go.

Members are invited to send in any questions they may have relating to the machine or games. No question is too big, no question too trivial. Ask it and we shall try our very best to answer it, or we promise to eat our joystick adapters. Questions which are sent in will be printed along with the answer (space permitting). Any question that cannot be answered by me or any of the team will also be printed (space permitting again) with the member's address (if they so desire) so that any other member who can answer the question can send the answer direct to the person. (It may help members to get in touch with others and share ideas). Who knows, maybe someone has the solution to quitting Eddie The Exterminator !

And before I forget, could you please remember to put your names and addresses on all documents that are sent in. Due to the hectic life I lead papers are getting separated and I find it hard to work out who sent in what. So if Fred Smith does not want his tip for getting past screen 1 in Devil's Lair credited to Joe Bloggs then please, please, please (beg, beg) put your name and address on all documents.

Below are a few tips and questions we have received from members wishing to share their new-found techniques with the rest of the world (!!)...

We have had members writing in about how to use the 'TIME' and 'DATE' commands as they seem to have been cryptically written in the User Manual (and this mag, so I hear). Well, I shall have an attempt at explaining them.

First the format which is as below:-

```
DATE "YYYYMMDD" (ENTER)

Where 'YYYY'=Year
  "     'MM'=Month
  "     'DD'=Day
```

To enter the date 25th Dec 1999 you would type in the line as follows:-

```
DATE "19991225" (ENTER)
```

To print the date at any time, just use 'PRINT DATE$' (ENTER) and the current date will be printed as below.

```
19991225
```

With time it is slightly different :

```
TIME "HH:MM:SS"   (ENTER)

Where 'HH'=Hours  (24 hour format)
  "   'MM'=Minutes
  "   'SS'=Seconds
```

So, 10:25pm would be entered like this :-

```
TIME "22:25:00" (ENTER)
```

To print the time again as below :-

```
PRINT TIME$ (ENTER)
```

Someone out there is having trouble with the 'BIAS' command, as to what it actually does !

The 'BIAS' command is only used when in 16 colour graphics mode (hires or lores). What the command actually does is to assign the colours 8 to 15 in the palette to a group of eight colours. There are 32 groups of colours and they are arranged as below.

0-7 8-15 16-23 24-31 32-39 etc..

The numbers above are the actual colour numbers. So if you were to set the bias to number 34, then from above we can see that the group with the colours 32 to 39 would be chosen and the colours 8 to 15 in the palette would be assigned to these colours, ie.

Colour number 8 would = 32
Colour number 9 would = 33
and so on.

Mark Procter from Ruislip, Middlesex, writes to say that he has a better way of defining your own default text page than that described in Issue 3 :

1. Open and display your text page where you want.
2. SET EDITOR VIDEO channel number of your text page.
3. CLOSE £0.
4. OPEN £0:"EDITOR:"

NOTE: IN STEP 2, DO NOT PRECEDE CHANNEL NUMBER WITH '£', ie.

type SET EDITOR VIDEO 110.

Steps 1 and 2 can be performed in immediate mode, but 3 and 4 MUST be done from within a program.

Another tip from Mark, this time for disc users with IS-DOS.

1. Boot up IS-DOS

2. SAVE 0 ^.COM

Then just type '^' to run the command.

AND TIPS EXTRAVAGANZA

The above command has the effect of resetting to IS-DOS pointers to the Transient Program Area (TPA) and the previous program that had been called from disk under IS-DOS will be re-started.

NOTE: The machine will crash if no program has been called since booting IS-DOS. Certain programs eg. (CHKDSK) will corrupt the TPA and when you try to reactivate by typing '^' (ENTER) a corrupted version will run and even if the disk is OK, it will return errors.

Who says we're not world famous ?

Here is a routine which I have found very useful. This has been written by Frank Jagler from Hattingen, Germany.

Most Disk Operating Systems (DOS) provide what is termed an EXECute file. These files (there can only be one in existence on any directory) are booted up (run as soon as the computer is powered up) and then run. What they do is make the computer run through a set sequence, substituting all input expected from the user setting up certain parameters (screens, type of printer, date etc.)

EXDOS does provide such a file (useful if you have EXDOS and a disc drive) but have no fear, Mr. Jagler does provide such a system for tape based users.

The listing provides two procedures, one for creating the file and the other for reading it. After having typed 'RUN', the procedures can be started by entering 'CALL CREATE' and 'CALL READ'.

IS-BASIC handles all communications through an editor channel (£0). The Editor in turn uses the Keyboard device for input and the Video device for output. All that has to be done is to 'Capture' the input from the tape channel and substitute it for the input expected from the keyboard channel (£105). This is what READ does.

After entering the program type 'RUN'. You will be asked to enter a command, just enter the command as you would normally do in immediate mode, eg. LIST, TEXT, RUN etc..

When you have finished entering the commands just enter 'XXX' to exit. The program creates a file called 'EXECUTE'. This is the file that the user will run each time the computer is used.

```
100 PROGRAM "BOOT.BAS"
110 DEF CREATE
120 TEXT
130 OPEN £10:"EXECUTE" ACCESS OUTPUT
140 DO
150 LINE INPUT PROMPT "COMMAND: ":A$
160 A$=UCASE$(A$)
170 IF A$="XXX" THEN EXIT DO
180 PRINT £10:A$
190 LOOP
200 PRINT £10:"CLOSE £10"
210 END DEF
220 !
230 DEF READ
240 OPEN £10:"EXECUTE"
250 CAPTURE FROM £10 TO £105
260 END DEF
270 END
```

Now for some tips, hints and problems playing games.

User having trouble with Return To Eden. Would like the answers to these problems. How to get past the Leviathian, what are the Airbrush and the Fig Leaf for? Also how to get off the Tiny Island. Doesn't ask much does he?.

Well, come on guys and gals surely there is a solution out there to help this poor lost soul (I know... tell you next issue.. heh heh - Ed).

Please can anybody send in tips for playing Airwolf and Devil's Lair which must surely rate as two of the hardest games in the universe, especially Devil's Lair. This game seems to be made to test our very sanity.

I have only one small tip for Airwolf and that is to try and shoot the small boxes with the small light-blue tip as these will remove some of the barriers. I'm still playing the game so hopefully more tips to follow.

Someone has written in asking how to make the Hidden Surfaces demo (Issue 4) work on the 64. Well, as the owner of said machine, what I have done is to make the variable 'NFRAMES' in line 240 equal to 10. This seems to work just fine.

Staying with the same program, it looks like the Basic doesn't work with the line:

```
IF COL > 0 THEN
```

This line is numbered as 770 on the Greatest Hits tape or as 850 in Issue 4.

If you change the line to :

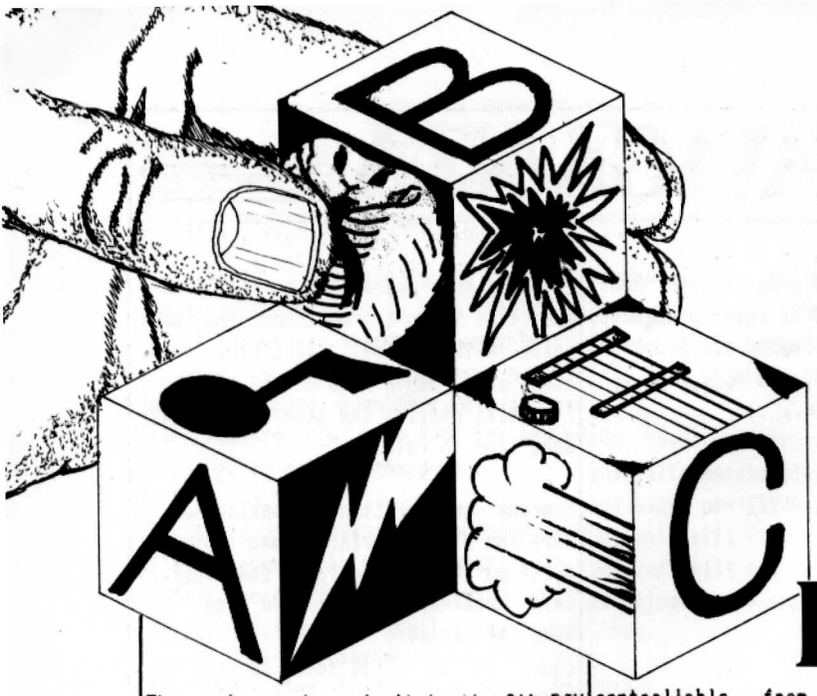
```
IF SGN (COL) = -1 THEN
```

the program works ok.

One last point - someone has sent in the solutions to rescuing all eight sorcerors in the game Sorcery. This, I feel, would ruin the playability of the game so I shall not print them here, but anyone can send in a request for them if they so wish and a copy will be forwarded to them.

So now it is up to you to start sending in those tips and questions (the more the merrier). Please do not wait for Joe next door to send the tips because if you all do that, guess what happens !! BI BI for now ...

Mike Turner



THE PEASY GUIDE TO USING SOUND ON THE ENTERPRISE

Those of you who made it to the 9th PCW Show may have had the (mis)fortune to come across the guys of the IEUG blasting out of a 100 watt Amplifier various 'orrible noises. I have to admit they were my fault. I also have to admit I was lucky in missing this as they were attempting to get their mouse to work at the time (Friday), and trying not to jump on it in rage and frustration.

Anyway, enough of the grovelling apologies, as to make amends, I have produced something of a 'first' in 'Private Enterprise' - an article by a person not called Dave Race?? No, an article on how to use the 'Dave' sound chip, written by An Idiot, for, ahem, Non-Technically Minded Persons such as the majority of the IEUG membership.

To kick off, the 'Dave' custom sound chip can be described as being able to output four-part sound. The first three channels or sources give out a clear square-wave tone. These are numbered 0 to 2. The fourth source is the obligatory 'white noise' channel, which supplies a hissing or 'slushy' noise, as with an untuned radio. It is known as source 3 (Look at the manual...).

This summary could describe the standard arrangement of most computer sound chip capabilities (Amstrad et al..). The Enterprise 'Dave' chip departs from this in a number of interesting ways, in that it is able to offer such options as genuine stereo, controllable from both BASIC and ENVELOPE statements. Also,Ring Modulation, which lends a metallic 'Spacey' effect to any sound. There are several types of white noise, and a high and low pass filter. These are directly accessible from BASIC in the form of the SOUND STYLE command. Listed below are a few well known ones (and some less well known ones I discovered..)

SOUND STYLE 16 - Low distortion. This can, under certain circumstances give rise to a sound not unlike an electric guitar. It can be combined with other SOUND STYLE options from STYLE 64.

SOUND STYLE 32 - Medium distortion. A compromise between STYLE 16 and 48. I personally have not much used it. Like STYLE 16, can be combined with other STYLE parameters.

SOUND STYLE 48 - High distortion. Gives a thunder-like noise, and unlike white noise, is a relatively 'clean' sound and can be varied by pitch, thus making it more flexible in use. Can again be combined with other STYLE parameters.

SOUND STYLE 64 - High pass filter. This is most commonly used on two sound channels together (0 and 1) to produce the filter effect in which only the part of a tone above the filter cutoff point is heard,to give a high 'strained' quality to a sound. Can be combined with all the previous STYLE parameters and some others.

SOUND STYLE 128 - Ring modulation. This is achieved with channels 0 and 2 working together to produce that distinctive 'spacey' metallic sound described earlier. Variants of this STYLE are discussed below.

SOUND STYLE 144 - A fateful union of SOUND STYLES 128 and 16, use as STYLE 128. Also note STYLEs 160 and 176, which add ring modulation to STYLE 32 and 48 respectively.

SOUND STYLE 192 - A combination of ring modulation and high pass filter. Again use as STYLE 128. This option leads to a very subdued sound on its own. Beef it up with, say something in another channel or STYLE...

SOUND STYLE 255 - the highest number possible. Gives rise to a musical, tinkling white noise sound type. It is possibly an amalgam of all previous STYLES?? SOUND STYLEs 250-255 are best used either on their own, or else in channel 3 with STYLE 128 in channel 1.

N.B. What I think is the LOW PASS FILTER, is activated by SOUND STYLE 34 in source 3 (the white noise channel), in conjunction with various STYLES, in source 1. Doing as the manual (at least my early incorrect copy) suggests, that is, STYLE 32 in source 3, with A.N. Other STYLE in source 2, does not the blindest bit of good as there is too much unwanted white noise...

In order to illustrate the above information, here is a brief easy BASIC listing. In this case, the extra dimension of ring modulation is added to what would be a continuous rise and fall STYLE 48 peal of thunder.

```
100 DO
110  FOR C=1 TO 3
120   FOR A=125 TO 1 STEP-1
130    SOUND PITCH A,DURATION 4,
       SOURCE 2,STYLE 176
140  NEXT A
150    FOR B=125 TO 1 STEP-2
160     SOUND PITCH B,DURATION 4,
        SOURCE 0,STYLE 176
170    IF C=3 THEN STOP
180    NEXT B
190  NEXT C
200 LOOP UNTIL C=3
```

A few points to note; DO.. LOOP becomes DO.. LOOP UNTIL, with the controlling variable as C, and the control point as line 170. If you want to include this routine in a larger sound or music program, line 170 could become a GOTO (yuk ! Ed.) statement. DO.. LOOP on its own would merely mean the program going on all night. It is the interaction of ring modulation on the two STYLE 48 sounds that produces the alarming electric 'ripple' of lightning discharge to break up what would otherwise be the single sound type in the program.

Here is another favourite routine of mine, a 'Heavy Metal' ascending guitar riff, produced by a combination of STYLE 192 in source 0, STYLE 64 in source 1,and STYLE 16 in 2 :

```
100 DO
110  FOR C=1 TO 3
120    FOR A=18 TO 46 STEP 2.4
130      FOR B=10 TO 20 STEP 2
140     SOUND PITCH A,DURATION B,
        STYLE 192,SOURCE 0
150     SOUND PITCH A-9,DURATION B,
        STYLE 64,SOURCE 1
160     SOUND PITCH A+12,DURATION B,
        STYLE 16,SOURCE 2
170     IF C=3 THEN STOP
180     NEXT
190   NEXT
200  NEXT
210 LOOP UNTIL C=3
```

Again note that DO.. LOOP is the main controlling structure of this routine, and that the STOP statement on line 170 can be replaced with GOTO (ugh! another one! Ed.), if you wish to include it in any larger programs.

A major part of the Enterprise's sound production capabilities is the ability to modify the pure, boring square wave tone by the use of ENVELOPE in a SOUND statement. ENVELOPE is set up, firstly by giving it a number and line, eg.

```
100 ENVELOPE NUMBER 1;2,0,40,30;5,20,
    63,20;1,-63,63,10;6,63,-63,50
```

The ENVELOPE can be broken down into four parameters per phase. The first number sets the pitch value (High or Low?), the second and third values control the LEFT and RIGHT stereo channels respectively. Finally, the last number before the semicolon deals with duration (50=1 second). The Enterprise ENVELOPE is flexible enough to permit anything up to 21 phases (Generated from Sound Utility in Issue 3.) of these four values, or else they can be kept as simple as you want. You are not, on the one hand, stuck with setting up 14 parameters before you get a squeak out of the BBC ENVELOPE. Or stuck with the Commodore'sASDR limitation. However, as in the example above, the ASDR (Attack,Substain,Decay, Release) philosophy is best to stick with if you want an easy life.....

To the example ENVELOPE,add these lines.....

```
110 SOUND PITCH 40,DURATION 150,
    STYLE 48,ENVELOPE 1
120 SOUND PITCH 45,DURATION 200,
    STYLE 48,ENVELOPE 1,SOURCE 1
```

The ENVELOPE modifies the sound by creating a left to right 'Wraparound' effect, controlled by the stereo parameters, particularly in the last two phases. This particular ENVELOPE had a total duration of 110, or just over 2 seconds. Generally, short sharp (ie. twangy) sounds are better produced with lower duration values, and longer, more 'whooshy' sounds need high duration values.

The usefulness of the ENVELOPE statement is only limited by your imagination. For example, this listing below works equally well with or without an ENVELOPE, but the sound in each case is quite different....

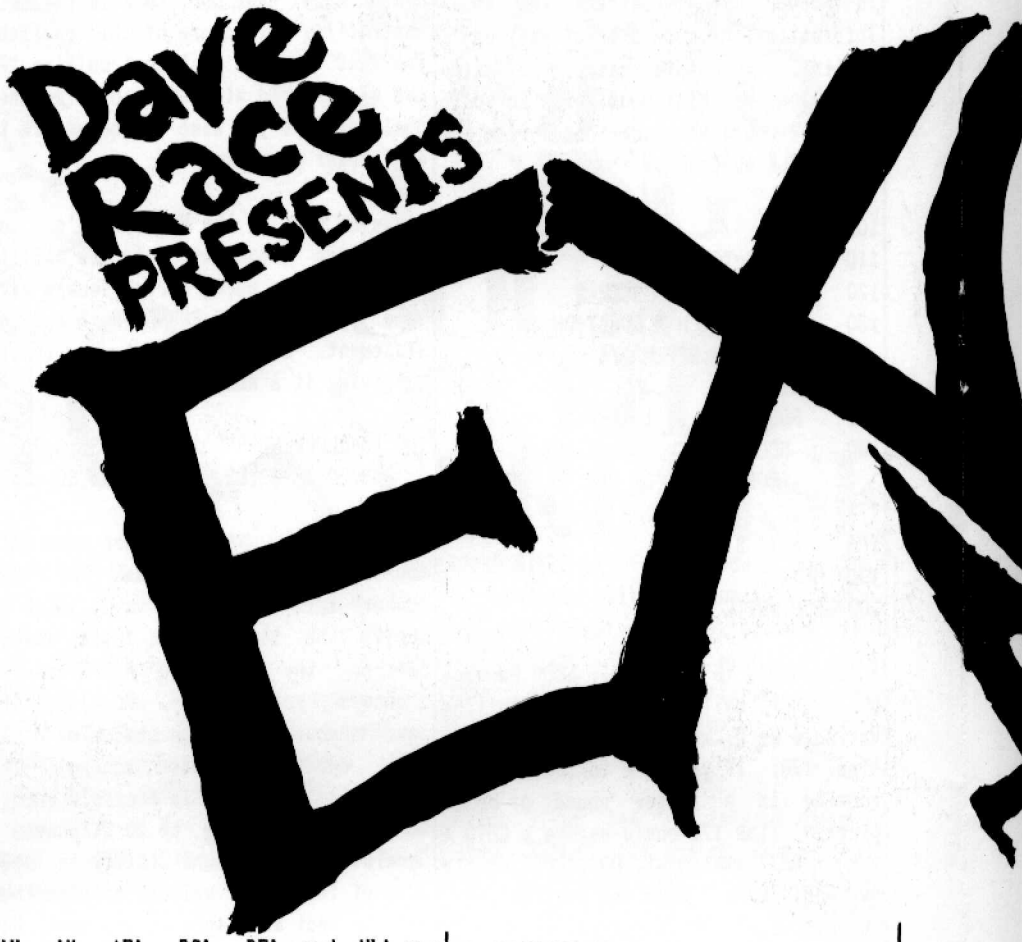
```
100 ENVELOPE NUMBER 1;-12,63,20,5;2,40,
    10,4;3,30,63,8
110  FOR A=57 TO 58 STEP .2
120   FOR B=80 TO 40 STEP-3
130    SOUND PITCH A+B+B,DURATION 9,
       STYLE 128
140    SOUND PITCH A/B/B,DURATION 9,
       SOURCE 2,STYLE 128,ENVELOPE 1
150   NEXT
160  NEXT
```

To notice the differences, remove line 100, and also the ENVELOPE statement on the end of line 140. In this case it is also interesting to note that the ENVELOPE is only three phases long, that is twelve parameters in total.

Finally, ("Thank God!" I hear you cry) there are a number of areas and ideas I would like to touch on that more advanced users could try. For example, you could set up sound sources by OPENing and CLOSEing channels. The default Sound output channel is channel £103. The use of user-defined sound sources could put a greater flexibility of sound use into the users hands. I have not looked at the use of such areas as the SYNC statement, which sychronises sounds from more than one source, or the INTERRUPT statement, which can replace any current sound with a new one. These will need a different article. Other ways of expanding the sound and musical capabilities of the Enterprise include hardware modifications to enable MIDI connection to a keyboard or sequencer. This has been done (I am reliably informed by Tim Box...) by R.A.Penfold in his publication 'MIDI and your Micro' (Bernard Babini publishing) though I have yet to find the book.....Also, there is the possibility of adding a sound sampling capability with an analogue to digital convertor, and the appropriate software. I am sure there is someone out there who is putting the finishing touches to such a device...

Chris Holland

Private Enterprise publications proudly presents EXOS Function Calls , or The Enterprise Technical Manual in 144 weekly parts.



If anyone can remember back to issue 7 they will recall that I did an article on system variables. In the article I made the promise that we would be covering EXOS function calls in the near future, i.e. sometime in the next year or so. What I didn't realise was that our beloved editor would think such an article would be a great idea, especially if yours truly wrote it. So here I sit, Neil standing over me with a sawn off Doc Martin, desperately trying to come to terms with machine code .... again !

Because I am not exactly one of the worlds greatest machine code programmers, anyone want to buy a slightly used assembler, I shall try to present the subject of EXOS function calls in a way that will hopefully be of use to, and understood by, Basic programmers. Purists should be warned that this will involve calling machine code routines from basic programmes and the use of the infamous ALLOCATE command.

In general a function call is made by executing RST 30h followed by the number of the function, 'h' indicates that the number is in hexadecimal e.g. 30h is the same as 48 decimal. The function may require that certain registers are set before it is called, these are referred to as it's parameters and will be explained for each function call.

Usually the function will give values to certain registers, the results of the function call. Register A, for instance, is normally used as an error indicater. If everything has gone O.K. A will be zero, otherwise it will be set to a value indicating what sort of error has occured, the Z flag will also be reset to 1 if an error has occured. If a RST 18h is performed after a function call an error can be detected and the machine will return to basic with a suitable error code. Unless stated otherwise the register pairs HL, IX, IY, AF', BC', DE' and HL' are preserved by EXOS function codes.

Take a look at the article on system variables, there is an example of function 16 which is used to set, read or toggle a system variable. The code shown is a little meaningless as I have put the value 3 in B, which is supposed to be set to 0, 1 or 2, so let's change it to something a little more sensible:

```
LD B,2      ‹toggle the variable›
LD C,26     ‹the variable controlling
             whether the status line is
             displayed›
RST 48      ‹calls a EXOS function›
DEFB 16     ‹specifies which function is
             required›
RST 24      ‹check to see if an error
             has occured›
RET         ‹return to wherever the
             routine was called from›
```

This will toggle the status line every time the code is called, and could be coded with the following basic program:

```
10 ALLOCATE 10
20 CODE TOGGLE_STATUS=HEX$("06,02,0E,
   1A,F7,10,DF,C9")
30 CALL USR(TOGGLE_STATUS,0)
```

Which when run will toggle the status line - magic, eh? Of course this would mean that you would need three bits of code for every system variable, which does sem a little wasteful. Note, however that CALL USR allows a value to be passed to HL; this value could then be used to decide which variable is being affected, and in the case of SET the desired value could be passed to D. The resulting code would look like this.

```
LD C,H      ‹the value of H will be set
             by the call routine›
LD D,L      ‹again the value will be set
             by the call routine›
LD B,1      ‹this time we want to set
             the variable›
RST 48      ‹          ›
DEFB 16     ‹all as above›
RST 24      ‹          ›
RET         ‹          ›
```

In basic this would be:

```
10 ALLOCATE 10
20 CODE SET_VARIABLE=HEX$("4C,55,06,
   01,F7,10,DF,C9")
```

The function would then be called by the command:

```
CALL USR(SET_VARIABLE,256*VAR+VAL)
```

where VAR is the system variable being set and VAL is the value to be given to the variable, e.g.

```
CALL USR(SET_VARIABLE,7167)
```

will set the border white.

As you have probably realised by now anyone who wants to use these special function calls is going to have to be able to convert the mnemonics to hex values, and a little knowledge of machine code would be a great help. The tome I use is "Programming the Z80", by Rodnay Zaks. Whilst the teaching part of this can be a little heavy it does have a tremendous reference section on the Z80 instruction set and I heartily recommend it.

There is a complete list of function calls on pages 201 and 202 of the programming guide, as you can see many of them are concerned with handling channels, so I shall deal with these functions first of all - yes, this will be another of my multipart epics.

Function 1 Open Channel

Parameters:

A Number to be given to channel. This must not be set to 255, this is the channel used to cancel redirects and captures - more about which later.

DE Address in memory of the device and/or filename that this channel will be associated with. The first byte will be the length of the string.

The device name takes the form:

```
{{device-name}{"-"}unit-number}":"}
```

where {} denotes an optional part and " " denotes literal characters. The device name can be up to 28 letters long. The unit number, if specified, can be seperated from the device name with "-" or it can immediately follow it.

Examples: TAPE:, VIDEO:, DISK-1: etc.

The filename can also be up to 28 characters long, and follows the usual rules of EXOS / EXDOS filenames. If no device name is given then the default device will be used.

If using Basic, then code the device / filename seperately. This will allow you to obtain the address from the variable used in the code statement, don't panic! an example follows the description.

Results:

A Error indicator, will be non-zero if an error has occured during the function call. See the beginning of this article for more details.

Description:

This function opens a channel just like the basic OPEN command does. If an attempt is made to open a file that does not already exist then an error will be generated.

Unit numbers and filenames are ignored by devices that do not support them, e.g. VIDEO:,KEYBOARD: etc. If a unit number is not given and the device does support unit numbers, e.g. DISK:, then the default will be used, the current drive in the case of DISK:.

It should be noted that many devices require that certain system variables are set before a channel is opened to them, just as they would if a channel was opened from basic, e.g. before

openning a channel to VIDEO: system variables 22 to 25 must be set to the desired values.

Example.

Suppose we wanted to open channel 1 to a file that we have already created, say TEXTFILE.WP for example. The machine code for this would be:

```
LD A,1
LD DE,FILENAME ADDRESS
RST 30h
DEFB 1
RST 18h
RET
```

In Basic this would be:

```
 10 ALLOCATE 100
 20 CODE FILENAME=HEX$("0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0,0,0")
 30 LET FILENAME$="TAPE:TEXTFILE.WP"
 40 POKE FILENAME,LEN(FILENAME$)
 50 FOR COUNT=1 TO LEN(FILENAME$)
 60   POKE FILENAME+N,ORD(FILENAME$(N))
 70 NEXT COUNT
 80 CODE OPENCHAN=HEX$("3E,01,11")&
    WORD$(FILENAME)&HEX$("F7,01,DF,C7")
 90 CALL USR(OPENCHAN,0)
100 FOR COUNT=1 TO 100
110   GET £1:A$
120   PRINT A$;
130 NEXT COUNT
140 CLOSE £1
```

Line 10 simply allocates an area 100 bytes long to be used for machine code, and moves the basic program up 100 bytes in memory; plenty enough for our program.

Line 20 fills the first 18 bytes of this with zero. We will put the file name in this area. The CODE command puts the memory address at which this area starts into the variable FILENAME.

Line 30 puts our filename into the string variable FILENAME$.

Line 40 pokes the length of our filename into the memory location held in FILENAME, then lines 50 to 70 poke the ascii values of each character in the filename into the memory. We now have the filename in memory, with it's length byte at the beginning, at the address stored in the variable FILENAME.

Line 80 puts the machine code routine in memory, the command WORD$(FILENAME) converts the address held in FILENAME into the correct hex format. The mnemonic:

```
LD DE,FILENAME ADDRESS
```

becomes:

```
HEX$("11")&WORD$(FILENAME)
```

Line 90 calls the machine code routine, and lines 100 to 140 simply demonstrate that the routine has worked. If you want to prove that the error handling works in the routine leave out the CLOSE command in the last line and try running the program twice. You should get an error message the second time.

Function 2 Create Channel

Parameters: As for function 1 above.

Results: As for function 1.

Description:

This function is identical in every way to function 1 except for the cassette and disc drivers.

In the case of the cassette driver this function operates in the same way as the basic command:-

```
OPEN £chan:"filename" ACCESS OUTPUT
```

whereas function 1 is equivalent to:-

```
OPEN £chan:"filename" ACCESS INPUT
```

With the disk driver this function will open a file, destroying any existing file of the same name. Obviously the file does not have to already exist for this function to work.

In both of the above cases the device being used will try to set aside a buffer for input/output, if there isn't enough RAM for this buffer an error will be generated.

Function 3 Close Channel

Parameters:

A Channel number. As usual this should not be 255.

Results:

A Error indicator, as in function 1.

Description:

This simply closes the specified channel, just like the basic CLOSE command. An error will occur if an attempt is made to close a channel that does not exist.

Before the channel is closed it's buffers are flushed and any RAM allocated to it is freed for use.

For example if we were dealing with a tape channel there would be an area of RAM, the channel buffer, in which characters would be stored before being sent to the tape. The computer will wait until this buffer is full and will then send it's contents to the tape s a single block. Obviously the chances are that when you close a channel there will be some characters in it; flushing means that these characters are sent to the tape, as opposed to losing them. When the buffer has been flushed the RAM set aside for it will be de-allocated.

Function 4 Destroy Channel

Parameters: As for function 3.

Results: As for function 3.

Description:

This function is identical to function 3 for all devices except the disk driver.

In the case of the disk driver this function will delete the file being accessed by the specified channel.

Function 5 Read Character

Parameters:

A Channel number. Not 255

Results:

A Error indicator, as in function 1.

B ASCII value of character returned from channel.

Description:

This function reads a character from the specified channel and puts it's ASCII value in B.

The function will wait until a character is ready, thus it is possible for a routine to hang up indefinitely when using this function. When dealing with files an error will be generated if an attempt is made to read a character after the end of the file has been reached.

Example.

The following code will read a character from the keyboard, channel 105, and put its ASCII value in register L which will then be returned to the user.

```
LD A,105  <The keyboard channel>
RST 30h   <Call function   >
DEFB 05   <    and         >
RST 18h   <check for errors>
LD H,00   <Zero register H and >
LD L,B    <put value in B into L>
RET
```

This becomes the basic program:

```
10 ALLOCATE 100
20 CODE READ_CHR=HEX$("3E,69,F7,05,DF,
   26,00,68,C9")
30 DO
40   PRINT CHR$(USR(READ_CHR,0))
50 LOOP
```

Line 40 prints out the character defined by the value in L. The program will continue until either the stop key is pressed or until an invalid code is sent to the editor.

In fact the code to clear register H is not necessary as the USR function puts zero into the register pair HL when the code is called. This means that we can leave out the code 26,00 in line 20.

Function 6 Read Block

Parameters:

A Channel number to read from. Not 255.

BC Number of bytes, i.e. characters to be read. This should of course be coded in the usual way, i.e. least significant byte followed by most significant byte.

DE Buffer address, i.e. the address in memory where the characters will be stored.

Results:

A Error indicator.

BC The number of bytes left to be read. This will only be of importance if there has been an error during the read.

DE The modified buffer address. This will be equal to the original buffer address plus the number of bytes read and therefore follows the data which has just been read.

Description:

This function reads a block of data, the size of which is specified in BC, from the specified channel. As the length of the block is stored in BC it may be anything from 0 to 65535 bytes long.

The first character read in will be put at the address specified in DE, any further characters will be stored sequentially from this point; for the interest of machine codists this block may cross segment boundaries.

An error will be generated if an attempt is made to read past the end of a file, although data up to the end will have been read and stored correctly.

As mentioned above, BC and DE will be valid even if an error occurs so that the read may continue from the byte where the error occured.

Example:

The following code will use functions 1, 3 and 6 to open a channel to a previously created file, TESTFILE.TXT; read a block of data 1000 bytes long from the file; store the data at address 10000 onwards; and finally close the channel:

```
LD A,1
LD DE,FILENAME ADDRESS
RST 30h
DEFB 1
RST 18h

LD A,1
LD BC,1000
LD DE,10000
RST 30h
DEFB 6
RST 18h

LD A,1
RST 30h
DEFB 3
RST 18h
RET
```

The different function calls have been seperated to make the code easier to understand. As you can see we have to load A with the channel number after every function call. This is because A is used as the error indicator and so its value is changed by the function calls. The Basic for this code would be:

```
100 ALLOCATE 200
110 CODE FILENAME=HEX$("0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,0,0")
120 LET FILENAME$="TESTFILE.TXT"
130 POKE FILENAME,LEN(FILENAME$)
140 FOR N=1 TO LEN(FILENAME$)
150   POKE FILENAME+N,ORD(FILENAME$(N))
160 NEXT
170 CODE PROG=HEX$("3E,01,11")&
    WORD$(FILENAME)&HEX$("F7,1,DF")
180 CODE READBLK=HEX$("3E,01,01,E8,03,
    11,10,27,F7,6,DF")
190 CODE CLOSE=HEX$("3E,01,F7,3,DF,C9")
200 CALL USR(PROG,0)
210 FOR N=10000 TO 11000
220   PRINT CHR$(PEEK(N));
230 NEXT
```

Again the seperate functions are shown as three different code statements.

When the first block is called, CALL USR(PROG,0), all three blocks will be run through until the RET is reached in the third block.

Lines 210 to 230 are there to prove that the data has been read correctly.

Function 7 Write Character

Parameters:

A Channel number to write to. Not 255

B ASCII code of character to be written.

Results:

A Error indicator.

Desription:

This function sends the character stored in B to the specified channel.

An error will be generated if the channel has not been openned.

If writing to a file the character will be placed in the channel buffer, if this fills the buffer, its contents will be written to the file. This is why a channel's buffer is flushed before the channel is closed.

Function 8 Write Block

Parameters:

A Channel number to be written to. Not 255.

BC Number of bytes to be written.

DE Address where characters are to be read from.

Results:

A Error indicator.

BC The number of bytes that are still to be written. This will be valid even if an error has occurred.

DE The modified buffer address, i.e. the address that the write function has got up to.

Description:

This function is the 'write' equivalent of Function 6, read block. It will send a block of data, the length of which is stored in BC, starting at the address in DE to the channel specified in A.

As in Function 7 a channel buffer will be used when writing to a file.

BC and DE are valid at all times, as in Function 6, so if an error occurs operation can be continued from the byte which caused the error.

Example.

In the example for Function 6 I have used a FOR...NEXT loop to peek the values read in from the text file. This could be replaced by the following code:

```
LD A,0        (We want to send the data)
LD BC,1000    (to the Editor, channel 0)
LD DE,10000
RST 30h
DEFB 8
RST 18h
RET
```

Note that BC and DE are given the same values as they were for function 6.

In the Basic program we would replace lines 210 to 230 with:

```
210 CODE WRITEBLK=HEX$("3E,00,01,E8,03,
    11,10,27,F7,08,DF,C9")
220 CALL USR(WRITEBLK,0)
```

Function 9 Read Channel Status

Parameters:

A Channel number, not 255.

Results:

A Error indicator.

C Channel status, this will be one of the following:

00h if a character is ready to be read,
FFh if at the end of a file,
01h if neither of the above.

Description.

This function checks to see whether a character can be read from the channel specified in A.

It is especially useful if reading from a non-file handling device where a read character call would wait indefinately if a character was not available.

Example.

The following code would be part of a larger program, the code checks to see if a character is waiting to be read from the keyboard. If there is a character ready it is read, otherwise the read function is skipped:

```
LD  A,105 (Load A with the keyboard  )
          (channel                   )
RST 30h   (Check to see whether there )
DEFB 09   (is a character waiting to  )
RST 18H   (be read from channel 105   )
CP C      (Compare the value in A with)
          (that in C. If no error has)
          (occured  then A will be)
          (reset to zero at this point)
          (and so if C is also 0 the Z)
          (flag will be set.          )
JRNZ 6    (If the Z flag is not set)
          (then C wasn't 0, i.e. no)
          (character is waiting from)
          (the keyboard. So jump over)
          (the following call if Z is)
          (not set.                  )
LD A,105  (The keyboard channel again )
RST 30h   (Read a character. This will)
DEFB 05   (only be run if a character)
RST 18H   (is ready to be read.      )
...       (Continue with the rest of)
          (the program               )
```

Note that in the above it is not necessary to zero A before comparing it with C as this will already have been done by the previous function call, presuming no error occurred.

Next issue I shall continue with function calls pertaining to channels, with such goodies as redirecting and capturing channels, conning a disk machine that it wants to load from tape all the time and RANDOM ACCESS FILES !

See you in a couple of months.

Dave Race

The CP/M Spot

(Write to IEUG PUBLIC DOMAIN SOFTWARE at the User Group address enclosing a s.a.e. for a free synopsis catalogue.)

By Neil Blaber

When I wrote the first CP/M Page in the last issue I had no idea of the amount of interest it would generate. My thanks to everyone who sent in orders, even though it meant a few very late nights / early mornings trying to make up the backlog due to the sheer numbers involved !

CATALOGUES

Due to the frantic rush to get Issue 7 out, the material that appeared on the CP/M Page last time wasn't actually the final copy of what I wanted printing. One change that didn't make it into the mag involved clarifying an ambiguity about the catalogues on disk. The £5.00 fee doesn't actually cover the cost of the disk, although it does include return postage.

PRICES REVISION

Another change that didn't make it into the mag was a pricing revision !

The copying charge remains the same :

£2.50 per volume for 1-9
£1.75 per volume for 10+

However, my disk prices have dropped significantly (esp. 3.5") :

£1.50 per 5.25" (DS) disk for 1-9
£1.00 per 5.25" (DS) disk for 10+

£2.50 per 3.5" (DS) disk for 1-9
£2.00 per 3.5" (DS) disk for 10+

Postage and packing rates remain the same (50p for up to first 3, then 7p per disk after that).

Phew, that's all the bodges from December out of the way - now on to the new stuff...

YOUR CONTRIBUTIONS

In addition to putting tested and installed CP/M software in the IEUG library, I am also very interested in any Enterprise-specific software that anyone cares to put into the library. So, dig out all those old software projects that were shelved halfway through because they were unsaleable, and give this method of distribution a thought. If you think that you may still be able to make some money on it, donate it as a "User Supported" item for which you will receive a small sum if the purchaser finds your program of use. More details on this from me on request.

PUBLIC DOMAIN v USER SUPPORTED

Some of the items in the library are what is known as "User Supported" programs. This means the author retains copyright, although the Public Domain is used for distribution purposes. It is a "Try Before You Buy" system in which the user is encouraged to send the author a sum of money if he feels the program to be worthwhile. It is a moral obligation rather than a legal one, of course, to pay up if you use the program and think it is good. However, there is a problem in that most of the programs in the library are of American origin, and that sending money off to the States may seem to most of you to be rather a waste of time. I will endeavour to arrange things with specific authors if the demand is great enough (I have already had a number of people enquire about donating to the author of Leisure Ledger, reviewed later).

REVIEWS

Due to the sheer number of volumes involved, people have found it very difficult to pick out the better items from the library. To aid people in this, I will from this issue onwards review a number of programs which have attracted my particlur interest. Anyone wishing to write a short piece on any volume they have ordered (either which they think is excellent or that should be avoided at all costs) is invited to do so. The CP/M Spot may hopefully become a major part of the mag if you all write in ! Anyway, here's this issue's reviews.

SCRIVENER (Vol CPMBBUK 73)

Scrivener is a "macro text processor". It is unlike any text processing program I have ever seen in that it allows you to enumerate figures contained in an ordinary text file. This is only one of its many functions, however, and it also allows you mailmerging facilities (the ability to automatically put different names, etc. on standard letters, the names and addresses being held in another file). It comes with a huge, very well written manual (text file on disk), and is well worth investigating. It needs no installation, and will work with any ASCII file (WP files printed to disk are fine).

ZASM - PROLINK (Vols CPMBBUK 56/7)

If you're an assembly language programmer and are sick of DEVPAC, well maybe here's your salvation. ZASM is a fully implemented Z80 assembler which is also compatible with Microsoft's M80 assembler (and hence able to assemble any M80 code in the Library). Prolink is a Microsoft L80 compatible linker, and together they make a formidable team. ZASM has been fully trsted in the field and is currently being used to write a major piece of Enterprise-specific software.

LEISURE LEDGER (on Vol CPMBBUK 66)

This is a very easy to use, menu-driven personal ledger program. You can actually maintain 3 ledgers at once, and it achieves such functions as automatic debit of account on a particular date without the user having to learn a complex set of commands (such as is the case with spreadsheet programs such as SuperCalc). However, this is a User Supported program and is a "limited life" demo version, although it is fully implemented. I have written to the author in order to sort out the problem of payment if anyone thinks it is worthwhile (and I feel it certainly is). I have installed Leisure Ledger already for the Enterprise so it can be run straight away (although it has the best Install program I've seen yet in the Library).

HUNT THE WUMPUS (on Vol BOOG 11)

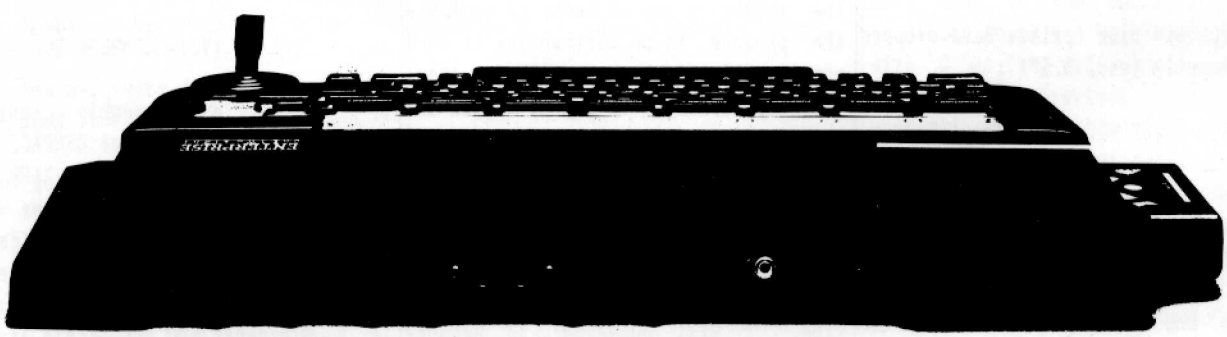
There are a vast number of games in the library written in Microsoft Basic. Almost without exception, they are so badly written that the coding makes Eddie the Exterminator look like a work of art. These are the games dredged up from the earliest days of microcomputers, before the advent of graphics or people who knew how to write games. They need MBasic to run, although there are ways round this (contact me for details), and have a morbid fascination that just makes you want to play more of them. Hunt the Wumpus is probably the worst game ever written in the entire history of computing. It is dull, totally random, requires no skill to play and is impossible to try to play properly anyway. I love it !

LATEST DEVELOPMENTS

Programs I am currently working on which should be available fairly soon in the IEUG library include WordPro-100, a nice word processor written in Turbo Pascal which I am altering and compiling (the source code is actually on the same volume as Leisure Ledger), so I can add features if there is a demand. Other programs being worked on include the C80 small C compiler (currently moded for the NewBrain), the Bradford fancy font program (although this is a struggle - it just seems to lock up the machine), and the PC-File database program (currently set up for the Osborne and has no Install program or easily recognisable data area).

Oh well, that's about it for this issue. Please let me have your comments on the volumes you order, as I don't get the time to look through all the stuff I'm sending out - it may well be that I'm overlooking some gems.

Neil Blaber



POWERHOUSE

require

FREELANCE PROGRAMMERS

For a Number of Projects Planned for 1987

We are concentrating on serious applications and games software, mainly disk–based, although certain programs will be released on tape if there is sufficient demand. If you feel you are a reasonably talented programmer/ designer, having a reasonable working knowledge of Z80 and/or a high–level language (such as C or Turbo/Hisoft/Pro–Pascal) and wish to gain optimum reward for your efforts, then contact Neil Blaber on

08926 – 63298

during office hours, or on

08926 – 3890

between 7.00pm and 10.00pm any day during the week.

If you have a program you are currently working on which you feel has the potential to be a hit, we would be delighted to discuss it with you.

Home Produce

```
100 PROGRAM "pal_anim.bas"
105 !*********************************
110 DIM CLR(9,8)
120 FOR X=0 TO 8
130   FOR Y=1 TO 7
140     READ CLR(X,Y)
150   NEXT X
160 NEXT Y
165 !*********************************
170 SET VIDEO COLOUR 2
180 SET VIDEO MODE 1
190 SET VIDEO Y 27
200 OPEN £101:"VIDEO:"
210 DISPLAY £101:AT 1 FROM 1 TO 27
220 LET INK=0:LET CN=0:LET CN1=2
230 !*********************************
240 FOR X=400 TO 971 STEP 4
250   LET CN=CN+1
260   IF CN>CN1 THEN LET CN=0:LET CN1=
      CN1*1.2:CALL INKINC
270   PLOT 0,X;1279,X
280 NEXT X
290 !*********************************
300 FOR X=971 TO 0 STEP-4
310   LET CN=CN+1
320   IF CN>8 THEN LET CN=0:CALL INKINC
330   PLOT 1279,971-X;1279-(X+100),0
340   PLOT 0,971-X;X+100,0
350 NEXT X
360 !*********************************
370 PLOT 640,260,
380 FOR X=1 TO 200 STEP 16
390   CALL INKINC
400   PLOT ELLIPSE X,200-X,
410 NEXT X
420 !*********************************
425 CALL PALSWITCH
426 !*********************************
430 DEF PALSWITCH
440   LET Z=2
450   DO
460     IF Z=1 THEN
470       LET Z=2
480     ELSE
490       LET Z=1
500     END IF
510     FOR X=0 TO 7
520       IF Z=1 THEN LET A=CLR(X,1)
530       IF Z=2 THEN LET A=255
540       LET B=CLR(X,2):LET C=CLR(X,3)
          :LET D=CLR(X,4):LET E=CLR(X,5)
          :LET F=CLR(X,6):LET G=CLR(X,7)
550       FOR Y=1 TO 20
560         SET PALETTE 0,G,A,B,C,D,E,F
570         SET PALETTE 0,F,G,A,B,C,D,E
580         SET PALETTE 0,E,F,G,A,B,C,D
590         SET PALETTE 0,D,E,F,G,A,B,C
600         SET PALETTE 0,C,D,E,F,G,A,B
610         SET PALETTE 0,B,C,D,E,F,G,A
620         SET PALETTE 0,A,B,C,D,E,F,G
630       NEXT Y
640     NEXT X
650     RESTORE
660   LOOP
670 END DEF
680 !*********************************
690 DEF INKINC
700   LET INK=INK+1
710   IF INK>7 THEN LET INK=1
720   SET INK INK
730 END DEF
740 DATA 64,8,72,1,65,9,73
750 DATA 64,136,200,17,81,153,217
760 DATA 128,80,208,10,138,90,218
770 DATA 128,16,144,2,130,18,146
780 DATA 192,24,216,3,195,27,219
790 DATA 128,48,176,6,134,54,182
800 DATA 0,32,32,4,4,36,36
810 DATA 64,40,104,5,69,45,109
820 DATA 192,56,248,7,199,63,255
```

TIM BOX & G. MORGAN

(FROME PAGEE 5E)E

companions for eating his pizza with his fingers. However, bribery was in the air, and for the price of a round of drinks Sir Spikey's life was spared.

And so it progressed, with the crowd being hypnotically drawn by the Bouncing Ball and the Megahertz while being repulsed by the bloated Lynx Beast (who still hadn't changed his shirt since the Show started) and other noxious creatures nearby, resulting in scores of people bouncing uncontrollably between the stands (and being goosed by the robot on the way). One exceedingly hairy specimen even interviewed Sir Garfield, although the magazine he worked for went Bang within minutes of misquoting his profound words. Sir Spikey and his trusty sidekick Sir Dataquip visited many a software house that day, and many business cards were exchanged for scruffy bits of paper due to a severe lack of IEUG business cards. Encouraging were the signs, and happy were the Knights at their progress. So happy were Sirs Spikey and Andy that they did partake of a binge with Palace Software and committed various atrocities in an Italian restaurant.

Chapter ye Fifthe - Drunkenness and a Near Miss

And verily did the sun rise to greet the Final Day, which saw Sir Andy (in his usual early morning state) and Sir Spikey (not much better) staggering sleepily towards the Castle. As the Castle became visible, a nearby motorist, blinded by Sir Spikey's hideous jumper, mounted the pavement and drove straight towards our heroes, pausing only to smash up the front of the local estate agents. Summoning up all their vast reserves of courage, the brave pair turned and legged it into the nearest alley, only just ahead of the oncoming car. The motorist, able to see once more, decided that stopping was a bad idea and accelerated up the road, trailing glass and tyre rubber behind him. The two, shaking their heads to clear the fog that such rapid movement had created between their ears, crawled on towards the Castle, unable to believe that they had almost been hit-and-run victims.

More sorties were made to software houses that day, but were unsuccessful due to copious amounts of intoxicating liquids having been imbibed by the subjects of the various negotiations. As the day progressed, more and more of the occupants of other stands were found either lying on the floor or in the process of running full-pelt to the loo while doing various facial exercises. Upon returning to the Stand, Sir Spiderman and Sir Andy were discovered to be in possession of various alcoholic substances and were seriously considering lying on the floor. After being force-fed half a jar of instant coffee granules each, they suddenly found themselves capable of carrying some of the baggage as the rest of the knights packed up the Stand.

Truly it had been a most successful campaign for the Knights, who then all dropped into total obscurity apart from Sir Andy who found a spare bottle on the way home and lived happily ever after.

THE END (FOR NOW)

Neil Blaber

Software Update

Name : SOEV Sound/Envelope utility
Producer : Boxsoft
Category : Utility
Price : £5.95

If you're one of those people who have been wanting to play about with the sound capabilities of the Enterprise, but have been dissuaded by the less-than-brilliant section in the Basic manual, then SOEV may be what you're looking for. It runs either as a system extension or as an applications program (you get both versions on your tape), and there are separate versions for the 64K and 128K machines (you in fact get all four on the tape). The first thing to greet you upon starting up SOEV is a yukky whoosing sound vaguely reminiscent of that associated with the early Entersoft offerings. However, things get much better from there onwards, as you are then plunged into the Sounds option, which incidentally is a wonderful yellow and brown affair.

You can then enter values for Sound Number, Envelope Number, Volume (Left and Right), Style, Source, Duration, Sync etc. and then press "E", which takes you into a screen where you can graphically define the shape of an envelope for the sound. You can then play the sound to find out what a mess you've made. You see, this is a purpose-built utility that can be used to muck about with, plugging in numbers until you get a sound you like, and then the resulting noise can be saved either as IS-BASIC code or as an escape sequence which can be used from Pascal, machine code or whatever.

If you felt lost when I started quoting the various categories of values to enter, not to worry - you don't need to know what they mean in order to make sounds with SOEV, and playing about is the best way of learning. SOEV fills what was quite an alarming gap in the range of Enterprise software - now let's see some sound demos appearing in the magazine !

Name : IS-BASIC Extensions
Producer : Boxsoft
Category : Language
Price : £7.95

If I told you that someone had produced a piece of software that would give you sprite control, a number of desirable but previously unavailable graphics functions and a built-in assembler as an addition to IS-BASIC and was selling it for under a tenner, you would think I was pulling one of your extremities. Well, my friends, I can say with hand on heart that such an item exists - it is Boxsoft's IS-BASIC Extensions. This inspirational artifact comes in five portions, which gives you the option of omitting a section containing commands or functions you are not using in a particular program. However, after the program has been written with the various parts loaded in a particular order, they must always be loaded in that same order for the program to run correctly (just make a note in a comment at the beginning of your program of which parts you used and in which order and you should be OK). I must admit that I didn't have the time to sit down and write my own example programs to test out the various functions that the Extensions provide, but having examined the example program that Boxsoft provide with the package and run it a number of times, my conclusion is that the sprite facility can be used to get good, fast onscreen graphics, even from raw uncompiled old IS-BASIC.

My complaints (and there are two) are that the documentation I was supplied with was concise to the point of only giving details of the syntax of the various commands together with a brief explanation, although Tim assures me that a new manual will be available about the time that you read this, and that anyone having bought the Basic Extensions prior to the new manual being written will be issued with one free when it is released. My other complaint, which is really only a quibble, is that there seems to be a bit too MUCH in the way of extra features, to the extent that some are a little extraneous. For example, how many times will you require a function built into a language which draws a menu on the screen which can only be a 40 character text screen with a maximum of 24 entries in the menu ? The section which does this can be conveniently left out, and all in all I feel IS-BASIC Extensions to be the best value for money of any utility yet released.

Neil Blaber

CONTINUED FROM P.7

commands Andrew Richards was to add. The assmebler built in to IS Extentions is only a simple one. If you want to write large and complicated amounts of code it is recommened you use a dedicated assembler.

THE ERRATA PAGE-WELL,ALMOST.

Issue 7 made history in many ways not only was it the biggest issue ever but it also had the largest number of errors. We would like to appologise for this state of affairs and put the major boobs to right here (for reasons of space we will not cover spelling errors or mistakes in the CP/M spot as this is delt with in detail in that section this issue).

Firstly the nameless writer of the piracy piece on the FORUM page was in fact Gary Thomson. In the middle of the second column on page 11 of the Enterprise saga the sentence should have gone'...it is not really worthwhile for enthusiasts, as opposed to buisness or specialist users to take a gamble with first generation 16-bit offerings; and this is doubly so for those starting out, who are not...'. The top line of the third column on page 18 of the Software register should read '...the presently unobtainable commercial software so far...'. Also the opening introduction to the Software archive page opposite this should read 'Despite the fact that most commercial software, with the exception of those reffered to on page 6, are at present...'. Finally the third line of the second column that reads on from this should continue '...facility (see opposite); starting...'.

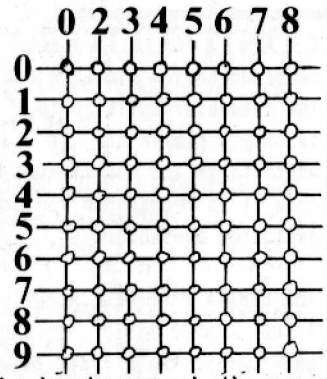
Advanced Programming

SCANNING THE KEYBOARD

Reading the keyboard is pretty fundamental to most programs. This is made simple in most cases by the use of the available EXOS functions. For those who want to detect more than one key being pressed, or who does not wish to use EXOS here is a direct way of reading the keyboard.

The keyboard is set up as a matrix of 10 by 8 lines, with each key at an intersection (see fig 1).

fig.1



To see if a key is pressed, the row on which that key lies must be selected. This is done by OUTing the row number to port £B5 (181). Reading port £B5 again gives which keys are pressed on that row. For each key there is a bit which is SET (1) if the key is NOT pressed and RESET (0) if pressed. The keys and the bits are shown in Fig 2. The external joysticks may also be scanned in this way but port £B6 (182) should be used for the IN command. The 0 bit is reset if the joystick function is true (see fig 3).

Writing an algorithm to test a key is fairly simple. The one problem is that you must not allow the normal EXOS keyboard scanning routine to run in between the OUT and IN commands, otherwise the port will be corrupted. This means that you must use machine code to disable interrupts before the OUT, and re-enable them after the IN. Those not using EXOS at all (EXOS being disabled) will have no problem.

To test for a key "A" pressed:

```
LD      A,1         ; "A" in row 1
DI                  ; disable interrupts
OUT     (£B5),A     ; Scan row
IN      A,(£B5)     ;
EI                  ; re-enable interrupts
BIT     6,A         ; test bit 6
JR      Z,yesitis   ; Yes A is pressed
..                  ; otherwise it's not
```

A general scanning routine can be simply written by combining the row and bit values for a key into a single number. The routine is given this number and returns a value showing if the key is pressed. One way of combining the values is to multiply the row number by 256 and to add the binary value of the bit to test. That is:

Bit 0 = 1 Bit 1 = 2 Bit 2 = 4
Bit 3 = 8 etc..

fig.2

BIT

ROW	0	1	2	3	4	5	6	7
0	N	;	B	C	V	X	Z	L-SHFT
1	H	LOCK	G	D	F	S	A	CTRL
2	U	Q	Y	R	T	E	W	TAB
3	7	1	6	4	5	3	2	ESC
4	F4	F8	F3	F6	F5	F7	F2	F1
5	8		9	"	0	^	ERASE	
6	J		K	+	L	*	]	
7	STOP	DOWN	RIGHT	UP	HOLD	LEFT		ALT
8	M	DEL	<	?	>	R-SHFT	ENTER	INS
9	I		0	@	P	[	SPACE	

fig.3

PORT B6
(External joysticks):

0 : 1-FIRE
1 : 1-UP
2 : 1-DOWN
3 : 1-LEFT
4 : 1-RIGHT
5 : 2-FIRE
6 : 2-UP
7 : 2-DOWN
8 : 2-LEFT
9 : 2-RIGHT

A Table of such values can be built up for direct use:

A = 320 B = 4 etc..

Here is the assembly listing of a general scanning routine for Basic.

```
; HL holds the information on the key
; to be tested :
; H=row number, L=bit mask
; HL returns 0 if key not pressed and 1
; if it is

iskey   LD      A,H
        DI
        OUT     (£B5),A
        IN      A,(£B5)
        EI
        AND     L
        JR      Z,yesitis
        LD      HL,0
        RET
yesitis LD      HL,1
        RET
```

The Basic Code for the routine is:

```
10 ALLOCATE 40
20 CODE ISKEY=HEX$("7C,F3,D3,B5,DB,B5,
   FB,A5,28,04,21,00,00,C9,21,01,00,
   C9")
30 !
40 DO
50 ! Test for Q (2+256*2)
60 QKEY=USR(ISKEY,514)
70 IF QKEY THEN PRINT "Q";
80 LOOP
```

In Pascal the routine is very similar, except that the IX register is used to access the parameters of the function.

```
FUNCTION iskey (keyno:INTEGER):BOOLEAN;
  BEGIN
    INLINE(£DD,£7E,£03,£F3,£D3,£B5,£FB,
    £DD,£A6,£02,£28,£04,£3E,0,£18,£02,
    £3E,1,£DD,£77,£04);
  END;
```

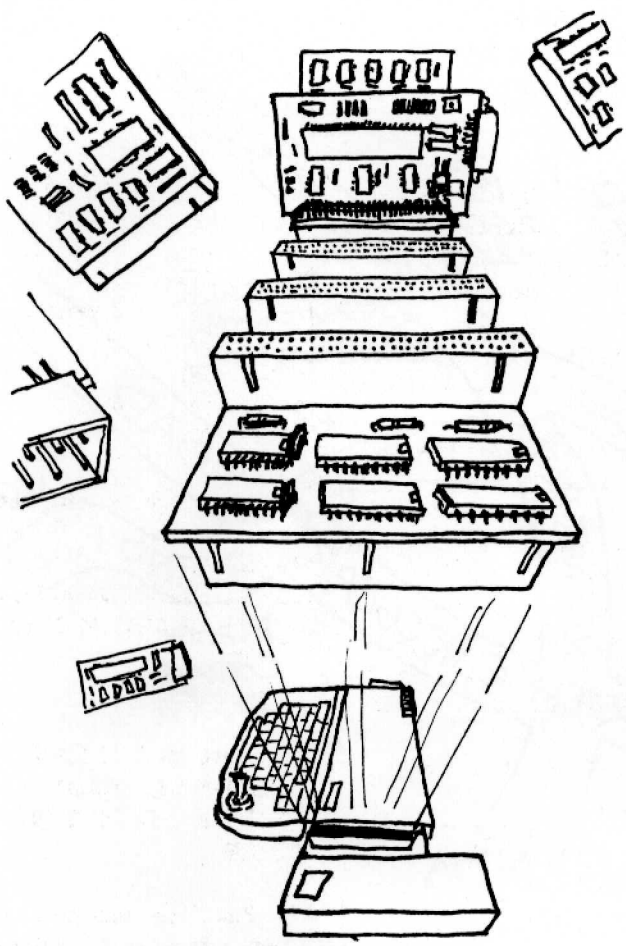
The function "iskey" returns TRUE if the key specified is pressed.

My thanks to Matthew Edwards for the port numbers and usage.

Gary Thomson

Announcing a major new step in Enterprise Expansion

Rememember the original Enterprise specification - "Expandable up to 4 Megabytes", and the superb original plans for it's hardware.



Soon you will be able to free your Enterprise from the limitations of it's memory and from the poor quality built-in ports, with a totally new expansion system, designed to grow with your computing needs.

A motherboard system is the backbone of a professional computer, and ours will take the Enterprise firmly into the realms of buisiness computers, and hardware and software development systems costing many times more.

A new range of plug-in cards is being developed, including memory cards; professional serial and parallel interfaces; relay control cards etc.; and exciting new developments like an EPROM programmer, sound sampler, video digitiser, second processor, and sprite board are in the pipeline for next year.

The basis of the system will be the "mini motherboard", very similar to the Enterprise - EXDOS connecting box, and sitting where the fabled "base unit" would have sat.

One card can plug into this mini motherboard, and will provide a low-cost introduction to the system. Larger motherboards will also plug into this mini motherboard, and will allow the connection of 4, 8 or more cards.

The system is compatible with all hardware expansions including EXDOS.

JOIN US!
We are a couple of electronic engineering students who are keen to see the Enterprise flourish as a top quality machine in the home and in business.

If you have experience in amateur or professional electronics design and construction, then we need YOUR help to get this system going.

Contact us if you have ideas for cards to fit on the system, or if you have designed and built similar circuits. We will pay generous royalties.

The motherboards should be in small-scale production by the end of the year, with lots of cards following on shortly.

Send an SAE for technical details to:

Andy Burnham,
Elvyn Richards Hall,
Ashby Road,
Loughborough,
LEICS.
LE11 3UQ

Dataquip Electronics

Expanding the Enterprise

Some estimated prices:

Mini Motherboard.....£15
4 slot motherboard.....£20
2 port parallel card...£20
256K Memory Expansion..£40

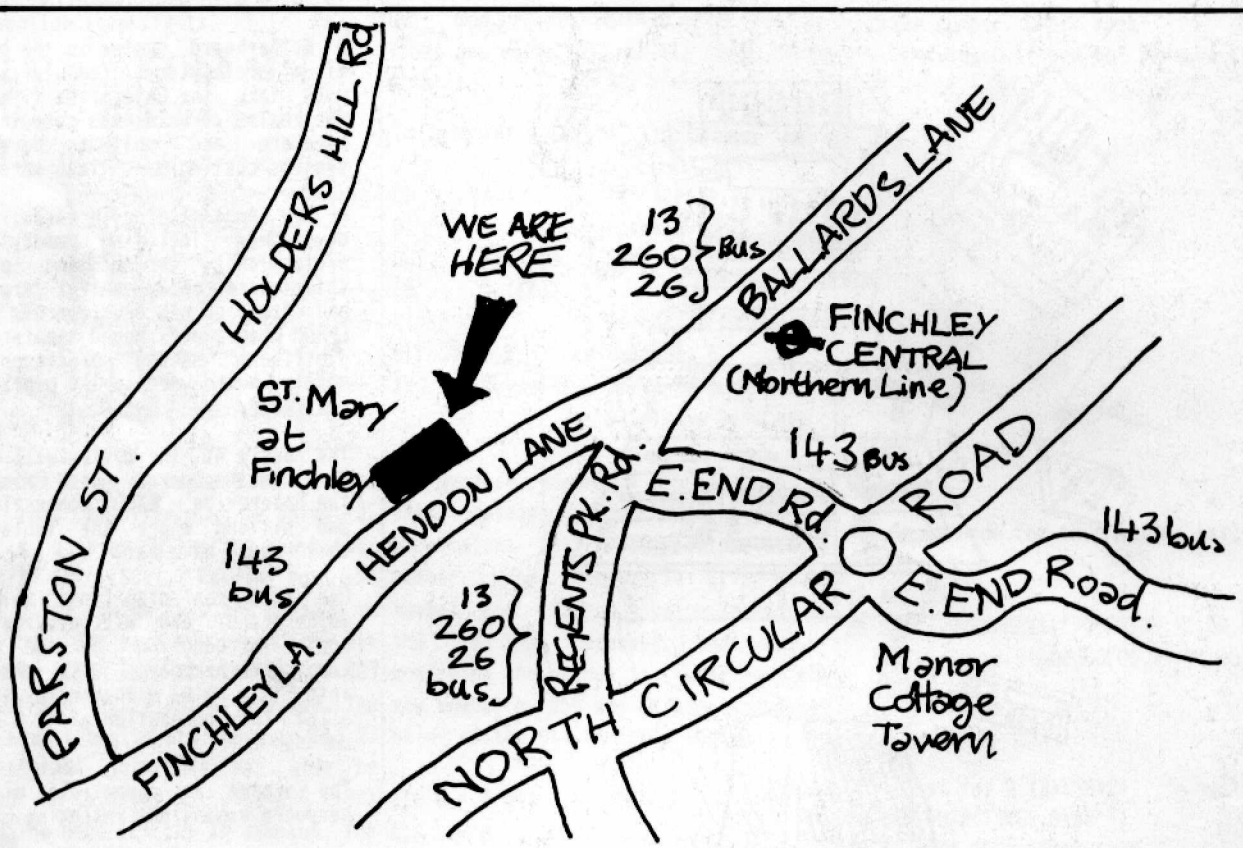
THE IEUG ANNUAL GENERAL MEETING 1986

which never actually happened on December 13th will now take place on :

Saturday, April 4th 1987 from 12.00pm - 5.00pm

at St. Mary's Hall, Hendon Lane, Finchley Central.

Bus routes are 143, 13, 260 and 26
Tube : Northern Line to Finchley Central.



AGENDA

1. Apologies for absence
2. Minutes of the Ordinary General Meeting held on Saturday December, 13th 1986.
3. Matters arising
4. Motion from the Executive.
   "That the proposals agreed in principle at the OGM held 13.12.86 andamended by Council be ratified forthwith. This Council meeting is now the Anuual Meeting of the IEUG."
5. Executive reports and questions from the Floor.
6. Election of Officers for period 1987/88.
   Chairman
   Secretary
   Treasurer
7. Financial strategy for 1987/88, including subscription fees for the forthcoming year.
8. Any Other Business.

Motions for "Any Other Business" and nominations to User Group posts, can be sent by post to the Secretary, IEUG, P.O. Box 13 Crowborough, EAST SUSSEX, TN6 1UL, or handed to him personally any time before 3pm on Saturday 4th April 1987.

Motions and nominations should be countersigned by a seconder in addition to being signed by the proposer, both of whom must be IEUG members. In the event of there only being a single candidate for any particular post, nominations may be taken from the Floor on the day of the meeting. Information on any particular post may be obtained from the Secretary at the above address or via the IEUG Hotline.